International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 7 (2019) pp. 1709-1714

Automatic Document Summarization Using Deep Learning Mechanism with Competent Analysis

Mythreagi. R[1*], Dr.N.Yuvaraj[2]

[1]UG Student, Department of CSE, KPRIEnT, Coimbatore-641407, India.

[2]Associate Professor, Department of CSE, KPRIEnT, Coimbatore-641407, India.

Abstract

Keywords play an important role in every aspect of search methodologies. Due to the rapid increase of information on the internet, keywords aid in for the effective search. Research on keyword mining is gaining more and more attention among researchers and learners. Document summarization generates a brief summary by extracting keywords from documents or from multiple documents by the deep learning algorithm. The main concept is to reduce the content or minimize the key important information present in any documents. The procedure is finagled by the Restricted Boltzmann Machine (RBM) algorithm for better efficiency and for removing redundant and unrelated sentences. The restricted Boltzmann machine is an undirected graphical model for binary random variables. It mainly consists of three layers input, hidden and the output layer. The input data is equally distributed to the hidden layer for further operation. The experimentation is carried out and the summary is generated for every document set given. From the summary, the document similarity is calculated from the selected inputs. It mains focus on the keyword iteration of each document given. This paper helps to calculate the accuracy of similarity among the documents.

Keywords: Deep Learning, Restricted Boltzmann Machine (RBM), Feature Extraction, Cosine Similarity, Rake Algorithm

I. INTRODUCTION:

For a decade most of the summarization is done in a manual manner. In the present time, the amount of information increases rapidly in very means over the internet and from every possible source. To overcome this issue, summarization is an essential way to tackle the overcrowding of information. Document summarization helps to maintain the text data by following some rules and regulations. Consider an example, the extraction of summary from a given documents shows a definite content from the whole document or multi-documents. Text summarization relates to the process of getting the textual document, processing the content from it and providing the necessary content in a shortened form. It should in a receptive way to the requirement of user or application. Normally text summarization can be classified in two ways, as abstractive summarization and extractive summarization. Natural Language Processing (NLP) technique is used for parsing, reduction of words and to generate text summary in abstractive summarization. Extractive summarization is said to be flexible and consumes minimum time when compared to abstractive summarization. In extractive summarization, it considers the sentence in a matrix form and with some basic feature vectors, the important sentences are obtained. A feature vector is said to be an n-dimensional vector with numerical features that represents the object. The objective of summarization is based on extraction approach which it selects the appropriate sentence as per the user's wish. Generally, text summarization is the process of reducing a given text content into a shorter version by keeping its main content intact and thus conveying the actual desired meaning. Single document summarization deals with a single document only[1]. Whereas multi-document summarization is the method of shortening, not with a single document, but with a collection of similar documents, in a single summary. This might look easy, but with implementation, it is a tough task to execute. Sometimes it may not be able to fulfill our desired goal. The degree of redundancy contained in a group of topically-related articles is considerably greater than the redundancy degree within an article since each article is appropriate to illustrate the most important point and also the required shared background. So, anti-redundancy methods play a vital role. The compression ratio might be less for a vast collection of related documents than for single document summary[2]. Text Analysis Conference (TAC) provides guided semantic information with important features. In this study, we have developed a multi-document summarization system using deep learning algorithm[3]. With the summary generated, it is trained with the help of Rake algorithm to generate the keywords. By the extracted keywords we can able to calculate the cosine similarity.

II. LITERATURE SURVEY:

i. Vishal Gupta and Gurpreet Singh

"A Survey of Text Summarization Extractive techniques". In this paper, the author describes the extractive summarization methods which comprise of two parts Pre Processing and Processing. In this paper, the pre-processing step is further divided into other subprocesses which are sentence segmentation, stop word removal and stemming. In the processing step, the weights are given to the features used for

extraction of summary from the large document respectively [4].

ii. Saranyamol C S and Sindhu L

"A Survey on Automatic Text Summarization." In this paper, the author describes the various techniques used in automatic text summarization which are extractive text summarization and abstractive text summarization[5].

iii. Rafael Ferreira, Luciano de Souza

Cabrala, Rafael Dueire Lins , Gabriel Pereira Silva , Fred Freitas, George D.C. Cavalcanti, Rinaldo Lima a, Steven J. Simske, Luciano Favaro, "Assessing sentence scoring techniques for extractive text summarization." This paper gives a brief description of various features used to perform extractive summarization and it also describes the methods for summary evaluation [6].

iv. K. Vimal Kumar, Divakar Yadav

"An Improvised Extractive Approach for Hindi Text Summarization." This paper mainly laid emphasis most importantly on the Hindi text summarization. It also describes various features used for the Hindi summarization using the extractive approach of text summarization. The author had proposed a system which can generate the summary with 85 % accuracy [7].

v. Vishal Gupta

"Hybrid Algorithm for Multilingual Summarization of Hindi and Punjabi Documents." The author of this paper has proposed a hybrid algorithm for Hindi and Punjabi text summarization. The algorithm proposed by the author is the first algorithm which can summarize both Hindi as well as Punjabi text [8].

vi. Ani Nenkova

"Summarization Evaluation for Text and Speech: Issues and Approaches." This paper suggests the methods for summary evaluation after the process of text summarization. Also, it describes some human models for summary evaluation [9].

vii. Inderjeet Mani "Summarization

Evaluation: An Overview." The author describes various methods for evaluating summary. [10].

III. IMPLEMENTATION PROCESS:

Restricted Boltzmann Machine:

Restricted Boltzmann Machine is a stochastic neural network that is a network of neurons where each neuron has some random behavior when activated. RBM has a single layer of visible units and one layer of hidden units. There is no intra_connection between the same layers. Connections between neurons are bidirectional and symmetric (fig-1). That shows the information flows in both directions during testing and training and while the usage of the network they hold the same weights in both directions. The flowchart represents the flow of execution for summarization (fig-2). It starts with the text document and preprocessing steps, which is later converted into the matrix form to create a summary.

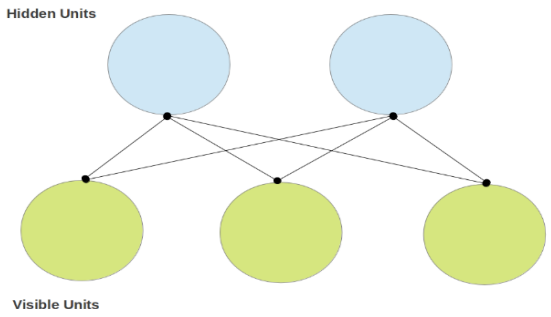

Fig 1: RBM network

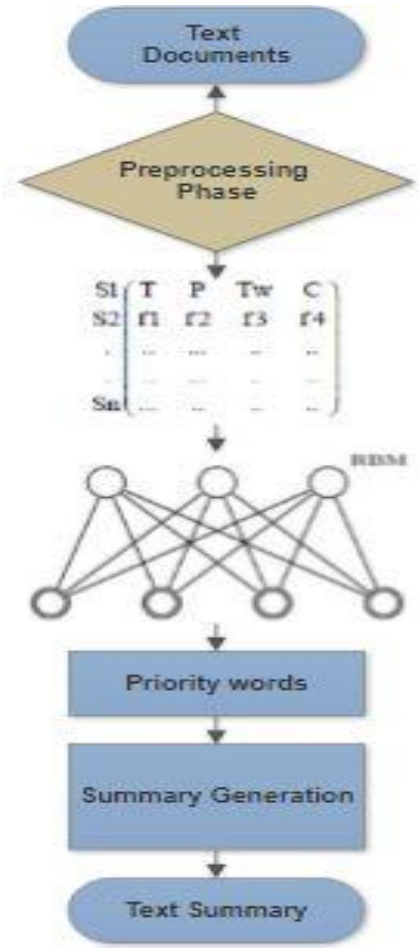

Fig 2: Block diagram for summarization

IV. NEEDED FEATURES

4.1. RBM Network Works in the Following Way

First, the network is trained by using some data set and setting the neurons on a visible layer to match data points in this data set. After the trained model the new unknown data is given to

make classification of the data which is known as the unsupervised learning.

4.2. Proposed Deep Learning Approach

Summarization technique is divided into two approaches extractive and abstractive. These methods are used for the selective field. Since there is a limitation in natural language generation techniques in generating the abstractive summary general extractive approach is prepared for summarization. For summarizing the text is structured into a certain model which is given as input to RBM. First, the text document is preprocessed using preprocessing techniques and later it is converted into sentence matrix. It is defined over a vocabulary of words in a file. The structured matrix of each row acts as an input to the model[11]. After getting the top priority words from the RBM the input query, sentence vector and high priority word output are compared to generate extractive summary for the document.

4.3. Preprocessing

To make the document light (not containing unwanted words) preprocessing of the text document for structuring is done by applying various techniques developed by the linguist. There are myriads of the technique by which we can reduce the density of the text document. In this study, we are using the following techniques.

- *Part of Speech Tagging*

Part of speech tagging is the process of marking or classifying the words of text on the basis of part of speech category (noun, verbs, adverb, adjectives) they belong. Varieties of algorithms are there to perform the POS tagging like hidden Markova models, using dynamic programming.

- *Stop Word Filtering*

Stop words are the words which are filtered out prior to or after the preprocessing task generally there is no specific rule on a particular word in a stop word, it completely depends on the text. In our condition, it is considered words like a, an, in by a stop word and it filters these words from the original document. Stop word filtering is the standard filtering in every text mining or text analysis applications.

- *Stemming*

Another important technique we need to apply is stemming. Stemming is a process of bringing the word to its base or root form, for example, using words singular form instead of using the plural (using boys as the boy), removing the ‘ing’ form verb (changing doing to do). There is a number of algorithms, generally referred to as stemmers', are there that can be used to perform the stemming.

4.4. Feature Vector Extraction

After the density of the document is reduced, the document is converted into a matrix. A sentence matrix S of order n*v is containing the features for every sentence of a matrix. For very informative summarization we are extracting four features of a sentence of text document via similarity with the title, relative position of the sentence, term weight of words forming sentences, concept-extraction of the sentence. Sentence matrix row vector represents the sentence which is making the document and the column vector contains the entry for these extracted features [12].

4.5. Term Weight

Term weight is a very important feature to consider for summarization. By term weight, it means the term frequency and its main importance. It is the most standard feature to be considered in natural language processing tasks. The frequency here is the term frequency which reflects the importance of a word in a document, it simply tells a number of times a word appears in the text. The term frequency of a word is given by tf(f,d) where f states the frequency of the word and d is the text document. The total term weight is calculated by computing by tf(f,d) and also the IDF for a document. The IDF refers to inverse document frequency which simply states whether the term is common or rare from all documents. It is calculated by dividing the total number of documents by the number of documents containing the term[13]. Later the log value is taken on the quotient. The IDF is given by:

$$idf(t, D) = log \frac{D}{d \epsilon D: t \epsilon d}$$

where, D is the total number of documents, $\epsilon D: t \epsilon d$ it is the number of documents where term t appears. The total term weight is calculated by

tf *IDF (t,d,D) =tf (t,d)* IDF (t,D}

f = tf *IDF .

4.6. Concept Feature

The concept feature from the text document is derived by either mutual information or windowing process. In the windowing process, a virtual window of size ‘k' is moved over a document from left to right. Here we want to find out the co-occurrence of words in the same window and it can be calculated by the following formula:

$$MI(w_i, w_j) = log2 \frac{P(w_i,\ w_j)}{P(w_{i)}) * P(w_j)}$$

where, $P(w_i, w_j)$ is the joint probability that both keywords appeared together in the window, $P(w_i)$-probability that a keyword w_i appears in a text window and can be computed by:

$$P(w_i) = \frac{|Sw_i|}{|Sw|}$$

Where

$|Sw_i|$= The number of windows containing the keyword w_i

|Sw| = Total number of windows constructed from a text document

4.7. Sentence Matrix

Here sentence matrix S = (s_1, s_2,……..s_n) where s_i=(f1, f2,……..f4), i<= n is the feature vector. The sentence matrix generates by the above steps is:

$$\begin{array}{c} \\ S1 \\ S2 \\ . \\ . \\ Sn \end{array}\begin{pmatrix} T & P & Tw & C \\ f1 & f2 & f3 & f4 \\ .. & ... & .. & .. \\ .. & ... & .. & .. \\ ... & .. & .. & .. \end{pmatrix}$$

4.8. Deep Learning Algorithm

The sentence matrix S = (s_1, s_2,……..s_n) which is the feature vector set having an element as s_i which is set contains the all the four features extracted for the sentence s_i. Here this set of feature vectors S will be given as input to the deep architecture of RBM as a visible layer[14]. Some random values are selected as bias H_i where i= 1,2 since an RBM can have at least two hidden layers. The whole process can be given by the following equation:

$$S = (s_1, s_2 \ldots\ldots s_n)$$

where, Si = (f1,f2,……..f4), i<= n where n is the number of sentences in the document. Restricted Boltzmann machine contains two hidden layers and for the two set of the bias value is selected namely H_0 H_1:

$$H_0 = \{h_0, h_1, h_2 \ldots\ldots h_n\}$$
$$H_1 = \{h_0, h_1, h_2 \ldots\ldots h_n\}$$

These set of bias values are values which are randomly selected. The whole operation of Sentence matrix is performed with these two sets of randomly selected value. The whole operation with RBM starts with giving the sentencing matrix as input. Here (s1,s2,……..sn) are given as input to RBM. The RBM generally have two hidden layers as we mentioned above. Two layers are sufficient for our kind of problem. To get a more refined set of sentence features. RBM works in a two-step. The input to the first step is our set of sentence matrix, S = (s1,s2,……..sn), which is having the four features of the sentence as an element of each sentence set[14. During the first cycle of RBM a new refined sentence matrix set:

$$s' = (s'_1, s'_2, \ldots\ldots s'_n)$$

The above-expressed s^l is generated by performing:

$$\sum_1^n s_i + h_i$$

During step 2 the same procedure will be applied to this obtained refined set to get the more refined sentence matrix set with H1 and which is given by:

$$s'' = (s''_1, s''_2, \ldots\ldots s''_n)$$

After obtaining the refined sentence matrix from the RBM it is further tested on a particular randomly generated threshold value for each feature we have calculated. For example, we select threshold thr_c as a threshold value for the extracted concept-feature. If for any sentence f4<thr then it will be filtered and will become a member of the new set of the feature vector.

Step 1-

[f1,f2,f3,f4] [f1,f2,f3,f4] [f1,f2,f3,f4]

$$\sum_1^n s_i + h_i(H_0)$$

$$s' = (s'_1, s'_2 \cdots, s'_n)$$

Step 2-

[f1,f2,f3,f4] [f1,f2,f3,f4] [f1,f2,f3,f4]

$$\sum_1^n s_i + h_i(H_1)$$

$$s'' = (s'_1, s'_2 \cdots, s'_n)$$

4.9. Summary Generation

In summary generation phase, the obtained optimal feature vector set is used to generate the extractive summary of the document. For a summary generation, the first task is obtaining the sentence score for each sentence of the document. Sentence score is obtained by finding the intersection of user query with the sentence. After this step ranking of the sentence is performed and the final set of sentences for text summary generation-defining the summary is obtained.

4.10. Sentence Score

Sentence score ratio is used to find the common words found in the user given data and a particular sentence to the total number of words in the text document. It is given by:

$$S_c = \frac{(s \cap Q)}{wc}$$

Where - Sc = Sentence score of a sentence, S = Sentence, Q = User query, Wc = Total word count of a text.

V. RAKE ALGORITHM

The RAKE algorithm is explained in a various way[16], but with the study, it is described as

- Candidates are extracted from the

text by finding strings of words that haven't form the phrase delimiters or stop words (a, the, of, etc). This produces the list of candidate keywords/phrases.

- A Co-occurrence form can be built to

identify the frequency that words are associated together in those phrases.

- A score is calculated for each phrase

that is the sum of the individual word's scores from the co-occurrence graph. An individual word score is calculated as the degree (number of times it appears + number of additional words it appears with) of a word divided by its frequency (number of times it appears), which weights towards longer phrases.

- Adjoining keywords are also

included if they occur more than thrice in the document and finds the score which is high enough. An adjoining keyword is said to be two keyword phrases which have a stop word between them.

- The top T keywords are then

extracted from the content, where T is 1/3rd of the number of words in the graph

- The cosine similarity is the cosine of

the angle between two vectors. In text analysis, each vector can represent a document. The greater the value of θ, the less the value of cos θ, thus the less the similarity between two documents.

In math equation:

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$

where cosine is the dot/scalar product of two vectors divided by the product of their Euclidean norms.

Table 1: The percentage of the similarity among the test documents

Test Documents	Cosine Similarity
Test 1 and Test 4	56.4%
Test 5 and Test 6	75.2%
Test 10 and Test 14	96.2%

VI. CONCLUSION AND FUTURE WORK

Several pieces of research were conducted for summary generation from the multiple documents. We have developed an automatic multi-document summarization system which incorporates the RBM as a single module. We have also used different features for the feature extraction phase. The feature score of the sentences is applied to the RMB in which the RBM rules are optimized with the help of Deep Learning Algorithm. The features are processed through different levels of the RBM algorithm and the text summary is generated accordingly. The generated result is tested as per the evaluation matrices. With the summary generated we have employed RAKE algorithm to find the frequently used keyword which is the second module. The experimentation of the proposed text summarization algorithm is carried out by considering various different document sets. The responses of test documents set to the proposed text summarization algorithm are satisfactory. The performance judging similarity values are 56.4%, 75.2%, and 96.2% respectively for the three sets document. The futuristic enhancement to the proposed approach can be done by considering different features and by adding more hidden layers to the RBM algorithm and improve the vector extraction directly into the application.

REFERENCES

[1] Darling, W.M. and F. Song, 2011. Probabilisticdocument modeling for syntax removal in text summarization. Proceedings of the 49th AnnualMeeting of the Association for computational linguistics, (CL' 11), ACM Press, Stroudsburg, PA., pp: 642-647.

[2] Goldstein, J., V. Mittal, J. Carbonell and M. Kantrowitz,Multi-document summarization by sentence extraction. Proceedings of the NAACL-ANLP Stroudsburg, PA, USA., pp: 40-48.DOI: 10.3115/1117575.1117580.

[3] M. Haque *et al.* "Literature Review of Automatic Multiple Documents Text Summarization", International Journal of Innovation and Applied Studies, Vol. 3, pp. 121-129, 2013.

[4] Vishal Gupta & Gurpreet Singh Lehal, "A Survey of Text Summarization Extractive Techniques", Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 3, August 2010.

[5] Saranyamol C S and Sindhu L, "A Survey on Automatic Text Summarization", International Journal of Computer Science and Information Technologies, Vol. 5(6), pp. 7889-7893, 2014.

[6] Rafael Ferreira et al. "Assessing Sentence Scoring Techniques for Extractive Text Summarization", Elsevier Ltd., Expert Systems with Applications 40 (2013) 5755-5764.

[7] Vimal Kumar K, Divakar Yadav "An Improvised Extractive Approach for Hindi Text Summarization" Springer India 2015

[8] Vishal Gupta," Hybrid Algorithm for Mul0tilingual Summarization of Hindi and Punjabi Documents" 2013 in Springer International publishing Switzerland 2013.

[9] Ani Nenkova, "Summarization Evaluation for Text and Speech: Issues and Approaches", Stanford University.

[10] Inderjeet Mani, "Summarization Evaluation: An Overview", USA.

[11] Kogilavani, A. and Balasubramanie P, 2012. Sentence annotation based enhanced semantic summary generation from multiple documents. Am. J. Applied Sci., 9: 1063-1070. DOI-2014.

[12] Mani, I., 2001a. Automatic Summarization. 1st Edn.,John Benjamins Publishing, Amsterdam, ISBN-10:9027249865, pp: 285.

[13] C.S.G. Khoo and D.H. Goh, 2008. Design and development of a concept-based multi-document summarization system for research abstracts. J. Inform. Sci., 34: 308-326.

[14] Patil, K. and Brazdil P, 2007. Text summarization: Using centrality in the Pathfinder network. Int. J. Comput. Sci. Inform. Syst., 2: 18-32.

[15] PadmaPriya, G. and K. Duraiswamy "An approach for text summarization Using deep learning algorithm", ISSN: 1549-3636, 2014.

[16] Michael W Berry, The Rake algorithm- book in Text Mining Applications and Theory.s